IN THE UNITED STATES PATENT AND TRADEMARK OFFICE

Serial No. 10/714,018	Conf. No. 1974
In Re Application of: Pizzoli *et al.*	Art Unit: 2191
Filed: 11/14/2003	Dkt. #: GB920030026US1 (IBML-0035)
	Examiner: Nguyen, Phillip H.

Title: SYSTEM AND METHOD FOR USER INTERFACE AUTOMATION

Mail Stop Appeal Brief- Patents
Commissioner for Patents
P.O. Box 1450
Alexandria, VA 22313-1450

BRIEF OF APPELLANTS

This is an appeal from the Final Rejection dated July 22, 2008 rejecting claims 1-20.

This Brief is accompanied by the requisite fee set forth in 37 C.F.R. 1.17(c).

REAL PARTY IN INTEREST

International Business Machines Corporation is the real party in interest.

RELATED APPEALS AND INTERFERENCES

Appellant is not currently aware of any prior or pending appeals, interferences or judicial proceedings which may directly affect or be directly affected by or have a bearing on the Board's decision in the pending appeal.

STATUS OF CLAIMS

As filed, this case included claims 1-20. Claims 1-20 remain pending. Claims 1-20 stand rejected and form the basis of this appeal.

STATUS OF AMENDMENTS

No amendment to the claims has been submitted subsequent to the Final Office Action dated July 22, 2008.

SUMMARY OF CLAIMED SUBJECT MATTER

The present invention provides a solution for language-neutral user interface automation. Below is a concise explanation of the subjected matter defined in the claims which are each involved in this Appeal. In addition, the summary points out elements in the figures that correspond to claim features as well as sections in the specification that discuss the features.

Claim 1 claims a system for language-neutral user interface automation, the system comprising: a system executing a test, the system executing the test including a processor; and a memory, (*see e.g.*, para. [0079]) the memory including: automation script means for receiving an automation script for automating use of the user interface (*see e.g.*, para. [0030], FIG. 4, elements 420, 450) in a system under test (*see e.g.*, FIG. 4, element 460) by the system executing the test, wherein the system under test includes an application (*see e.g.*, para. [0030], FIG. 4, element 460) and wherein the interface may be in an arbitrary natural language (*see e.g.*, para. [0038], FIG. 4, element 420, FIG. 6, generally); and script translation means for intercepting a call from the automation script to a function simulating a user action on the application; (*see e.g.*, paras. [0031]-[0033], FIG. 4, element 420, FIG. 5, element 510) wherein the interception includes

accessing a database or file system that is independent from the system under test so that the application's natural run-time execution is protected before, during and after the functional automation executes (*see e.g.*, paras. [0035]-[0041], [0072], FIG. 4, element 420, FIG. 5, element 510), retrieving a translated text string associated with the function call (*see e.g.*, paras. [0035], [0038], [0039]), and calling the function simulating the user action with the translated text string; (*see e.g.*, paras. [0038], [0039], [0041], FIG. 5, element 540) wherein the translation consists of converting to or from a first natural language to a second natural language (*see e.g.*, para. [0038], FIG. 6, generally).

Claim 2 claims the system of claim 1, wherein the script translation means comprises: message translation means for supplying translated text for the automation script's run time execution (*see e.g.*, para. [0032], FIG. 4, element 422); and selective text locator means coupled to the message translation means for selectively supplying appropriately translated text to the automation script's run time execution depending on the function call in a case that a same text string is translated differently based on context (*see e.g.*, para. [0032], [0036, [0060], FIG. 4, element 424).

Claim 7 claims the system of claim 1, wherein the automation script comprises a Java™ script. (*see e.g.*, para. [0042]).

Claim 10 claims a method for language-neutral user interface automation, the method comprising: providing an automation script for automating use of the user interface (*see e.g.*, para. [0030], FIG. 4, elements 420, 450) in a system under test (*see e.g.*, FIG. 4, element 460) by a system executing the test, wherein the system under test includes an application (*see e.g.*, para. [0030], FIG. 4, element 460) and wherein the interface may be in an arbitrary natural language (*see e.g.*, paras. [0038], FIG. 4, element 420, FIG. 6, generally); intercepting a call from the

automation script to a function simulating a user action on the application (*see e.g.*, paras. [0031]-[0033], FIG. 4, element 420, FIG. 5, element 510); wherein the interception is performed by the system executing the test and includes accessing a database or file system that is independent from the system under test so that the application's natural run-time execution is protected before, during and after the functional automation executes (*see e.g.*, paras. [0035]-[0041], [0072], FIG. 4, element 420, FIG. 5, element 510), retrieving a translated text string associated with the function call (*see e.g.*, paras. [0035], [0038], [0039]); wherein the translated text string has been converted to or from a first natural language to a second natural language (*see e.g.*, para. [0038], FIG. 6, generally); and calling the function simulating the user action with the translated text string (*see e.g.*, paras. [0038], [0039], [0041], FIG. 5, element 540).

Claim 11 claims the method of claim 10, further comprising: providing message translation means for supplying translated text for the automation script's run time execution (*see e.g.*, para. [0032], FIG. 4, element 422); and providing selective text locator means coupled to the message translation means, wherein the step of retrieving comprises selectively supplying appropriately translated text by the selective text locator means to the automation script's run time execution depending on the function call in a case that a same text string is translated differently based on context. (*see e.g.*, paras. [0032], [0036, [0060], FIG. 4, element 424).

Claim 19 claims a program product stored on a computer readable storage medium, for language-neutral user interface automation, comprising: program code for providing an automation script for automating use of the user interface (*see e.g.*, para. [0030], FIG. 4, elements 420, 450) in a system under test (*see e.g.*, FIG. 4, element 460) by a system executing the test, wherein the system under test includes an application (*see e.g.*, para. [0030], FIG. 4, element 460) and wherein the interface may be in an arbitrary natural language (*see e.g.*, para. [0038],

FIG. 4, element 420, FIG. 6, generally); program code for intercepting a call from the automation script to a function simulating a user action on the application (*see e.g.*, paras. [0031]-[0033], FIG. 4, element 420, FIG. 5, element 510); wherein the interception is performed by the system executing the test and includes accessing a database or file system that is independent from the system under test so that the application's natural run-time execution is protected before, during and after the functional automation executes (*see e.g.*, paras. [0035]-[0041], [0072], FIG. 4, element 420, FIG. 5, element 510); program code for retrieving a translated text string associated with the function call (*see e.g.*, paras. [0035], [0038], [0039]); wherein the translated text string has been converted to or from a first natural language to a second natural language (*see e.g.*, para. [0038], FIG. 6, generally); and program code for calling the function simulating the user action with the translated text string (*see e.g.*, paras. [0038], [0039], [0041], FIG. 5, element 540).

Claim 20 claims the program product of claim 19, further comprising program code for: providing message translation means for supplying translated text for the automation script's run time execution (*see e.g.*, para. [0032], FIG. 4, element 422); and providing selective text locator means coupled to the message translation means, wherein the step of retrieving comprises selectively supplying appropriately translated text by the selective text locator means to the automation script's run time execution depending on the function call in a case that a same text string is translated differently based on context. (*see e.g.*, paras. [0032], [0036, [0060], FIG. 4, element 424).

GROUNDS OF REJECTION TO BE REVIEWED ON APPEAL

I. Whether claims 1-20 are unpatentable under 35 U.S.C. 103(a) over Halviatti et al. (U.S. Pat. No. 5,475,843), in view of Straathof (U.S. Pat. No. 6,167,534).

II. Whether claim 7 is indefinite under 35 U.S.C. 112, second paragraph.

III. Whether claims 19 and 20 are directed to unpatentable subject matter under 35 U.S.C. 101.

ARGUMENT

I. Rejection of claims 1-20 under 35 U.S.C. 103(a) over Halviatti et al. (U.S. Pat. No. 5,475,843), in view of Straathof (U.S. Pat. No. 6,167,534).

In the Final Office Action, claims 1-20 were rejected under 35 U.S.C. 103(a) as allegedly being unpatentable over Halviatti et al. (U.S. Pat. No. 5,475,843), hereinafter "Halviatti" in view of Straathof (U.S. Pat. No. 6,167,534), hereinafter "Straathof." Applicants respectfully disagree with the Office's interpretation of the cited art. As discussed in more detail below, Applicants submit that the Office fails to show, *inter alia*, that the proposed combination of Halviatti and Straathof teaches or suggests a method, system or program product for language-neutral user interface automation that includes all the features claimed therein.

Interpreting Halviatti for purposes of this Appeal only, Applicants submit that Halviatti discloses a system and methods for automated program testing which requires predetermined, systematic and complex interception of the running application at run time. (*See e.g.*, Halviatti, at col. 8, lns. 48-67). The level of interception in Halviatti is intrusive, and has the disadvantage of creating a high probability of compromising the natural execution of the application. The methods used in Halviatti rely on an interception-based methodology or system, rather than allowing the application to run unhindered. (*See e.g.*, Halviatti, at col. 8, lns. 48-67, col. 10, lns. 43-48).

Moreover, the Halviatti system cannot intercept and control the resolution and management of complex user interface control types. In addition, in Halviatti's methodology, the application is centric, *i.e.*, the interception of all application messages, events (keyboard, mouse, system), states, etc., is a pre-requisite to success. (*See e.g.*, Halviatti, at col. 8, lns. 48-67). Such a model once again interferes with the natural execution of the application. Finally,

the methodology used in Halviatti encourages a level of abstraction which can potentially take some degree of control from the script writer. In this regard the script writer is dependent on the methodology and characteristics of Halviatti's interception and event handling engine. This limits the degree of control that the script writer can have in Halviatti's natural test abstraction development methodology. In summary, Halviatti's automated testing solution behaves in a way which encourages interference and interception of the running application. Halviatti relies on the specific interception of the running application to achieve its goals, and relies on the need to write ATUs (Application Translation Units). (*See e.g.*, Halviatti, at col. 8, ln. 48-67, col. 12, lns. 43-46).

A. <u>Rejection of Claim 1 under 35 U.S.C. 103(a) over Halviatti in view of Straathof</u>

Turning to claim 1, the Office argues that Halviatti discloses the claimed limitation: "wherein the translation consists of converting to or from a first natural language to a second natural language." Applicants respectfully disagree and submit that Halviatti does not teach this claimed feature. As discussed above, in direct contrast to the claimed invention, Halviatti's system is not concerned with language translation or anything to do with language or natural language. In Halviatti, the ATU traps events and translates them into abstract messages, or "meta-messages" for conveying information about a particular event to the system. (*See e.g.,* Col. 2, lns. 41-44). It does not translate between natural languages. Straathof also fails to teach such a translation to or from a first natural language to a second natural language. As such, Applicants submit that the Office has not shown that each and every claim limitation of claim 1 is disclosed by the cited references, and therefore Applicants submit that claim 1 is allowable.

In addition, with respect to claim 1, Applicants submit that the combination of Halviatti and Straathof fails to teach each and every element of the claims, including script translation means for intercepting a call from the automation script to a function simulating a user action on the application, wherein the interception includes accessing a database or file system that is independent from the system under test so that the application's natural run-time execution is protected before, during and after the functional automation executes. (*See* claim 1). As discussed above, Halviatti relies on the specific interception of the running application to achieve its goals, therefore not protecting the application's natural run-time execution, including the need to write ATUs (Application Translation Units), which are not necessary in the claimed invention. *See e.g.*:

> …the target application is registered with the Message Engine 350. In particular, hooks are installed by a corresponding ATU 340 so that events within the target application of interest are trapped.

(Halviatti, col. 10, lns. 43-47). Halviatti further teaches that each event is trapped for processing by an ATU. (*See e.g.*, Halviatti, col. 10, lns. 55-57.) This level of interception is intrusive and compromises the execution of the application at run-time. Installing a hook inherently changes an application and adds an extra step in a process, therefore affecting performance time. The hook functions between the application and the system, and therefore inherently changes the natural behavior of the application because the natural timing will be affected. (*See e.g.*, Halviatti, at col. 13, ln. 34-35, col. 8, lns. 48-67). Therefore, while the claimed invention protects the application's run-time execution, Halviatti specifically relies on interference during run-time.

With further respect to claim 1, the Office admits that Halviatti does not teach "accessing a database or file system that is independent from the system under test so that the application's

natural run-time execution is protected before, during and after the functional automation executes." Office Action, p.4. The Office cites to Straathof for allegedly teaching this claimed element. Again, Applicants disagree with the Office's interpretation of the cited art. Interpreting Straathof for purposes of this Appeal only, Applicants submit that Straathof teaches the use of a "Capture Agent" which captures one or both of the Windows and SQL API calls during a user session. (*See e.g.*, Straathof, col. 6, lns. 38-55.) The "Capture Agent" intercepts the user interface and application calls using a Windows API hook, for example. (*Id.*) As Straathof explains, this interception, as in Halviatti, also does not protect the application's run time execution: "After the "Capture Agent" intercepts the SQL calls, the "real" SQL call is then sent so that the user session may continue." (*See e.g.*, Straathof, col. 6, lns. 53-55.) Again, the hook used in Straathof functions between the application and the system, and therefore inherently changes the natural behavior of the application because the natural timing will be affected. Therefore, while the claimed invention protects the application's run-time execution, both Halviatti and Straathof specifically rely on interference during run-time.

With respect to dependent claims 2-9, Applicants herein incorporate the arguments presented above with respect to the corresponding independent claims from which the claims depend. Furthermore, Applicants submit that all dependant claims are allowable based on their own distinct features.

As such, Applicants submit that the Office has not shown that each and every claim limitation of claim 1 is taught or disclosed by the cited references, and therefore Applicants submit that claim 1 and claims 2-9, which depend therefrom, are allowable.

i. Rejection of Dependent Claim 2 under 35 U.S.C. 103(a) over Halviatti in view of Straathof

Furthermore, in addition to the discussion above, with respect to claim 2, Applicants submit that Halviatti further fails to teach, *inter alia*, selective text locator means coupled to the message translation means for selectively supplying appropriately translated text to the automation script's run time execution depending on the function call in a case that a same text string is translated differently based on context. (*See* Claim 2.) In support of its rejection, the Office asserts that Halviatti discloses that "'a message is retrieved by the call to GetMessage" and that "the retrieved message may be translated by a call to TranslateMessage." (Final Office Action, p. 6.) Applicants submit that a general call to translate a message in a Windows environment is not equivalent to selecting an accurate text translation from a plurality of available translations. In the present invention, the selective text locator ensures that during a search the correct translation is returned based on the type of control the textual object belongs to. The general translation call in Halviatti however, fails to teach this claimed feature. As such, Applicants submit that the Office has not shown that each and every claim limitation of claim 2 is disclosed by the cited references, and therefore Applicants submit that claim 2 is allowable.

B. Rejection of Claim 10 under 35 U.S.C. 103(a) over Halviatti in view of Straathof

With respect to claim 10, Applicants submit that the Office has failed, *inter alia*, to show that the proposed combination of Halviatti and Straathof teaches or suggests a method for language-neutral user interface automation that includes all the features claimed therein. For example, for reasons that should be clear from the discussion of the proposed combination of

Halviatti and Straathof above, Applicants submit that the proposed combination of Halviatti and Straathof fails to teach or suggest the method of claim 10, including "wherein the translated text string has been converted to or from a first natural language to a second natural language" and "intercepting a call from the automation script to a function simulating a user action on the application, wherein the interception is performed by the system executing the test and includes accessing a database or file system that is independent from the system under test so that the application's natural run-time execution is protected before, during and after the functional automation executes." As a result, Applicants respectfully submit that the rejection of claim 10 and claims 11-18, which depend therefrom, as allegedly being unpatentable over Halviatti and Straathof is improper.

i. Rejection of Dependent Claim 11 under 35 U.S.C. 103(a) over Halviatti in view of Straathof

With respect to claim 11, Applicants submit that the Office has failed, *inter alia*, to show that the proposed combination of Halviatti and Straathof teaches or suggests a method for language-neutral user interface automation that includes all the features claimed therein. For example, for reasons that should be clear from the discussion of the proposed combination of Halviatti and Straathof above, Applicants submit that the proposed combination of Halviatti and Straathof fails to teach or suggest the method of claim 11, including "providing selective text locator means coupled to the message translation means, wherein the step of retrieving comprises selectively supplying appropriately translated text by the selective text locator means to the automation script's run time execution depending on the function call in a case that a same text string is translated differently based on context." As such, Applicants submit that the Office has

not shown that each and every claim limitation of claim 11 is disclosed by the cited references, and therefore Applicants submit that claim 11 is allowable.

C. Rejection of Claim 19 under 35 U.S.C. 103(a) over Halviatti in view of Straathof

With respect to claim 19, Applicants submit that the Office has failed, inter alia, to show that the proposed combination of Halviatti and Straathof teaches or suggests a program product stored on a computer readable storage medium for language-neutral user interface automation that includes all the features claimed therein. For example, for reasons that should be clear from the discussion of the proposed combination of Halviatti and Straathof above, Applicants submit that the proposed combination of Halviatti and Straathof fails to teach or suggest the program product of claim 19, including "wherein the translated text string has been converted to or from a first natural language to a second natural language" and "program code for intercepting a call from the automation script to a function simulating a user action on the application, wherein the interception is performed by the system executing the test and includes accessing a database or file system that is independent from the system under test so that the application's natural run-time execution is protected before, during and after the functional automation executes." As a result, Applicants respectfully submit that the rejection of claim 19 and claim 20, which depends therefrom, as allegedly being unpatentable over Halviatti and Straathof is improper.

i. Rejection of Dependent Claim 20 under 35 U.S.C. 103(a) over Halviatti in view of Straathof

With respect to claim 20, Applicants submit that the Office has failed, *inter alia*, to show that the proposed combination of Halviatti and Straathof teaches or suggests a program product stored on a computer readable storage medium for language-neutral user interface automation that includes all the features claimed therein. For example, for reasons that should be clear from the discussion of the proposed combination of Halviatti and Straathof above, Applicants submit that the proposed combination of Halviatti and Straathof fails to teach or suggest the method of claim 20, including "providing selective text locator means coupled to the message translation means, wherein the step of retrieving comprises selectively supplying appropriately translated text by the selective text locator means to the automation script's run time execution depending on the function call in a case that a same text string is translated differently based on context." As a result, Applicants respectfully submit that the rejection of claim 20 as allegedly being unpatentable over Halviatti and Straathof is improper.

In summary, Appellants submit that claims 1-20 are allowable because the combination of Halviatti and Straathof fails to disclose each and every feature of the claimed inventions.

II. Rejection of claim 7 under 35 U.S.C. 112, second paragraph.

The Office has rejected claim 7 as allegedly being indefinite under 35 U.S.C. 112 because it contains the trade name JAVA. Applicants respectfully disagree. As the MPEP explains, names used in trade are permissible in patent applications if: (A) Their meanings are established by an accompanying definition which is sufficiently precise and definite to be made a part of a

claim, or (B) In this country, their meanings are well-known and satisfactorily defined in the literature. MPEP 608.01(v). JAVA is a well-known programming language that has been used for years and is well defined in the literature. Moreover, the term Java in claim 7 is not used in a source-identifying manner, instead it is used to describe the language in which the script is written. One of skill in the art would understand the meaning of the term Java in the claims and as such, Applicants submit that claim 7 is not indefinite.

III. Rejection of claims 19 and 20 under 35 U.S.C. 101.

The Office has rejected claims 19 and 20 under 35 U.S.C. §101 as allegedly being directed to non-statutory subject matter. Claim 19 is directed to a "program product stored on a computer readable storage medium." The Office cites to paragraph [0079] in the specification which states that "…the software may be provided as a computer program element carried on any suitable data carrier (also not shown) such as a magnetic or optical computer disc" to argue that claim 19 can include a data carrier. However, Applicants note that the claim language is specifically limited to a program product stored on a computer readable storage medium. To this extent, the claim is limited to data carriers that are computer readable storage media. A data signal, one type of data carrier, cannot store a program product. As such, Applicant has limited claims 19 and 20 to the Office's interpretation of statutory subject matter, and Applicants respectfully submit that the rejection of claims 19 and 20 under 35 U.S.C. 101 is improper.

Respectfully submitted,

/Meghan Q. Toner/

Meghan Q. Toner, Reg. No. 52,142
Hoffman Warnick LLC
75 State Street, 14th Floor
Albany, NY 12207
(518) 449-0044 - Telephone
(518) 449-0047 – Facsimile

Dated: December 9, 2008

CLAIMS APPENDIX

Claim Listing:

1. A system for language-neutral user interface automation, the system comprising:

a system executing a test, the system executing the test including a processor; and

a memory, the memory including:

automation script means for receiving an automation script for automating use of the user interface in a system under test by the system executing the test, wherein the system under test includes an application and wherein the interface may be in an arbitrary natural language; and

script translation means for intercepting a call from the automation script to a function simulating a user action on the application; wherein the interception includes accessing a database or file system that is independent from the system under test so that the application's natural run-time execution is protected before, during and after the functional automation executes, retrieving a translated text string associated with the function call, and calling the function simulating the user action with the translated text string;

wherein the translation consists of converting to or from a first natural language to a second natural language.

2. The system of claim 1, wherein the script translation means comprises:

message translation means for supplying translated text for the automation script's run time execution; and

selective text locator means coupled to the message translation means for selectively supplying appropriately translated text to the automation script's run time execution depending on the function call in a case that a same text string is translated differently based on context.

3. The system of claim 2, wherein the selective text locator means is arranged to selectively supply appropriate text to the automation script's run time execution depending on a resource ID of the function call.

4. The system of claim 1, wherein the script translation means comprises:

a library including a function having a same signature as the function call and which is arranged to retrieve the translated text string before the function call; and

one of:

a file referencing the library, the automation script being arranged to reference the file and the library, and

the library including the retrieval function and the function call, the library being arranged to be called by the automation script.

5. The system of claim 4, wherein the file referencing the library comprises an include file.

6. The system of claim 4, wherein the library including the retrieval function and the function call has the same name as a library containing the function called by the automation script.

7. The system of claim 1, wherein the automation script comprises a Java™ script.

8. The system of claim 1, wherein the automation script is in the English language and the application is arranged to use a non-English language.

9. The system of claim 1, wherein the user interface comprises a graphical user interface.

10. A method for language-neutral user interface automation, the method comprising:

providing an automation script for automating use of the user interface in a system under test by a system executing the test, wherein the system under test includes an application and wherein the interface may be in an arbitrary natural language;

intercepting a call from the automation script to a function simulating a user action on the application; wherein the interception is performed by the system executing the test and includes accessing a database or file system that is independent from the system under test so that the application's natural run-time execution is protected before, during and after the functional automation executes,

retrieving a translated text string associated with the function call; wherein the translated text string has been converted to or from a first natural language to a second natural language; and

calling the function simulating the user action with the translated text string.

11. The method of claim 10, further comprising:

providing message translation means for supplying translated text for the automation script's run time execution; and

providing selective text locator means coupled to the message translation means,

wherein the step of retrieving comprises selectively supplying appropriately translated text by the selective text locator means to the automation script's run time execution depending on the function call in a case that a same text string is translated differently based on context.

12. The method of claim 11, wherein the selective text locator means selectively supplies appropriate text to the automation script's run time execution depending on a resource ID of the function call.

13. The method of claim 10, wherein the script translation means comprises:

a library including a function having a same signature as the function call and which is called to retrieve the translated text string before the function call; and

one of:

a file referencing the library, the automation script referencing the file and the library, and

the library including the retrieval function and the function call, the library being called by the automation script.

14. The method of claim 13, wherein the file referencing the library comprises an include file.

15. The method of claim 13, wherein the library including the retrieval function and the function call has the same name as a library containing the function called by the automation script.

16. The method of claim 10, wherein the automation script comprises a Java™ script.

17. The method of claim 10, wherein the automation script uses the English language and the application uses a non-English language.

18. The method of claim 10, wherein the user interface comprises a graphical user interface.

19. A program product stored on a computer readable storage medium, for language-neutral user interface automation, comprising:

program code for providing an automation script for automating use of the user interface in a system under test by a system executing the test, wherein the system under test includes an application and wherein the interface may be in an arbitrary natural language;

program code for intercepting a call from the automation script to a function simulating a user action on the application; wherein the interception is performed by the system executing the test and includes accessing a database or file system that is independent from the system under test so that the application's natural run-time execution is protected before, during and after the functional automation executes;

program code for retrieving a translated text string associated with the function call; wherein the translated text string has been converted to or from a first natural language to a second natural language; and

program code for calling the function simulating the user action with the translated text string.

20. The program product of claim 19, further comprising program code for:

providing message translation means for supplying translated text for the automation script's run time execution; and

providing selective text locator means coupled to the message translation means,

wherein the step of retrieving comprises selectively supplying appropriately translated text by the selective text locator means to the automation script's run time execution depending on the function call in a case that a same text string is translated differently based on context.

EVIDENCE APPENDIX

No evidence has been entered and relied upon in the appeal.

RELATED PROCEEDINGS APPENDIX

No decisions rendered by a court or the Board in any proceeding are identified in the related appeals and interferences section.